An Experimental Study of Prediction Methods in Robust Optimization Over Time

Matthew Fox
School of Computer Science and Informatics, De Montfort University
Leicester, UK
matthew.fox@dmu.ac.uk

Shengxiang Yang
School of Computer Science and Informatics, De Montfort University
Leicester, UK
syang@dmu.ac.uk

Fabio Caraffini
School of Computer Science and Informatics, De Montfort University
Leicester, UK
fabio.caraffini@dmu.ac.uk

***Abstract*—Robust Optimization Over Time (ROOT) is a new method of solving Dynamic Optimization Problems in respect to choosing a robust solution, that would last over a number of environment changes, rather than the approach that chooses the optimal solution at every change. ROOT methods currently show that ROOT can be solved by predicting an individual fitness for a number of future environment changes. In this work, a benchmark problem based on the Modified Moving Peaks Benchmark (MMPB) is proposed that includes an attractor heuristic, that guides optima to a determined location in the environment, resulting in a more predictable optimum. We study a number of time series forecasting methods to test different prediction methods of future fitness values in a ROOT method. Four time series regression techniques are considered as the prediction method: Linear and Quadratic Regression, an Autoregressive model, and Support Vector Regression. We find that there is not much difference in choosing a simple Linear Regression to more advanced prediction methods. We also suggest that current benchmark problems that cannot be predicted will deceive the optimizer and ROOT framework as the peaks may move using a random walk. Results show an improvement in comparison with MMPB used in most ROOT studies.**

***Index Terms*—Particle Swarm Optimization, Computational Modelling, Metaheuristic Optimization, Benchmark Testing, Evolutionary Algorithm**

I. Introduction

Real-world optimization problems deal with changing objective functions, constraints and problem environments over a period of time [1]–[4]. These time-varied optimization problems that are known as Dynamic Optimization Problems (DOPs) require ad-hoc algorithmic solutions to continuously "chase" moving global optima. Other challenges include detecting when the problem changes and tracking the global optima. DOPs are conceived from the belief that learning from previous evaluations, when assuming the problem has not changed much, can increases the efficiency and reliability of the optimizer thus allowing for a quicker and more informed convergence to a solution.

A widely adopted approach of dealing with DOPs is tracking moving optima (TMO) on each environment change. Two strategies can be used to improve on TMO approaches: one is to use historical information [5] and the other is to maintain the population diversity [6]. However, TMO approaches do not take into account the cost of frequently changing solutions if the solution chosen is too costly or impossible to switch to.

A new approach that does take into account of these factors, called robust optimization over time (ROOT), was proposed by Yu [7] where the goal of optimization is to find consecutive solutions over time that can have a varied degree of optimality rather than finding the global optimum at each time interval. ROOT finds a solution that can be used over environment changes by maximizing the survival time of the robust solutions over time [8]. An acceptable optimal solution is problem specific and can be pre-defined by the user [9].

In this paper a novel experimental study is designed to test the performance of various time series forecasting methods. We test multiple regression techniques based on different paradigms, specifically; Linear and Quadratic Regression [10], a Autoregressive model [9], and Support Vector Regression (SVR) [11], to identify their success in finding a robust solution and establish whether a more accurate predictor achieves better results.

A new benchmark problem, is proposed that extends the MMPB problem by including a factor that attracts the peaks to a location. The latter allows for the movement to be predicted while keeping a satisfactory degree of random walk movement. We include an attractiveness weighting which attracts the peaks at different rates in order to influence the amount of random walk and noise in the predictive movement.

This paper is structured as follows. In Section 2 the related works of ROOT are reviewed. In Section 3, the current MMPB and new benchmark is proposed. The experiment methodology, performance measures and compared forecasting methods to be used are described in Section 4. Furthermore, in Section 5 the results of the experiments are discussed. Finally, Section 6 summarizes the findings and discuss potential further work.

II. Existing Research

ROOT was proposed as new method of solving DOPs in respect to choosing a robust solution that would last over a number of environment changes rather than the approach that chooses the optimal solution at every change [7]. A general framework was proposed by Jin [9] that consists of a population-based optimization algorithm (POA) as the optimizer; a database to store historical solutions information; a fitness-approximator and a fitness-predictor. In [9], a Radial Basis Function Network (RBFN) is adopted as the local approximator and an autoregressive (AR) model as the predictor. In further studies, the approximation methods were not used as a stronger focus was given on quantifying the effects of prediction error on the metrics to use the solution's true previous fitness instead [8], [12], [13]. The work in [14] studies the impact of various approximation models of past environments in ROOT considering the algorithm performance in terms of robustness. Precisely, both a Gaussian Radial Basis Function Network (referred to as RBFN-GAU) and Thin Plate Radial Basis Function (RBFNTHI) is included in the comparative analysis, as well as a Shepard Interpolation model (SHEPARD), a Support Vector Regression method equipped with a Gaussian Kernel (SVR-GAU) and one more equipped with a Laplacian Kernel (SVR-LAP). This experimentation unveiled that the number of peaks in the benchmark function directly impacts the performance of the approximation method. Moreover, although the obtained results suggest that selecting a suitable approximator significantly impacts the algorithm performance in ROOT, this is not enough to guarantee quality solutions close to the true optimum of the problem.

The suitability of ROOT methods compared to traditional TMO approaches remains an important characteristic that impacts the design of ROOT algorithms. The measurement of whether a ROOT approach is needed is vital and the incorrect use of ROOT may be deceptive in the problem and lead to worse results than expected.

Differently, the multi-swarm approach in [15] does not require an approximator and a predictor to base future solutions, as it only needs to work out metric values from the swarms to guide the search. This approach is extended in [16] to obtain a multi-swarm method capable of finding, tracking and monitoring peaks. In the latter, future behaviours of the peak are predicted based on information extracted from each swarm and exploited to pick the next robust solution when the current solution becomes unsatisfactory. Numerical results have shown that classic methods using survival times based metrics are less performing in highly dimensional problems as this task becomes almost impossible with larger search space and higher change frequencies.

Multi-objective approaches to ROOT have also been considered. In [17] the authors convert a scalar DOP to a two-layer multi-objective DOP to find the true robust solution set for both the survival time and average fitness simultaneously. In the first layer, the acceptable robust Pareto front considering both metrics is found at each time-varying moment. In the second layer, taking the average fitness and the length of the robust solution set as two objectives, the sequences of robust solutions chosen from the first layer during all time-varying moments are explored.

ROOT can also be extended to consider the switching cost from one solution to another solution as a metric to be minimized. This metric is termed Previous-Solution Displacement Restrictions (PSDR), or Switching Costs and forms of requirement satisfaction constraint in solving DOPs. PSDRs consider real-world problems where it may be expensive, or impossible, to change to a new solution if it is far from the current one in the problem space.

Typically, studies are focussed on the multi-objective approach to finding robust optimal solutions to DOPs and switching costs. [18] considers the switching cost, reflecting the degree of changes in implementing a new solution as an objective to be minimised while robustness is maximised simultaneously. In [19] the authors solves PSDRs by exploring the environment around the current solutions. This approach has a faster convergence time than TMO approaches thus being suitable for online DOPs. It is also worth mentioning the multi-objective Optimization of Adaptation method proposed in [20], which displays a hybrid structure implementing both ROOT and PSDR algorithmic behaviours. This work considers how the problem changes from the current optimum to the new optimum with a known location. The objective is to minimize the cost associated with the change between these solutions which is solved by creating a trajectory of solutions across the environment space to the new optima. The possible conflict between the optimality of the performance during the change and the cost of the adaptation defines a new optimization problem where the cost is minimized while the function value is maximised. This problem is addressed with an EA in [20] but other algorithms may also be used to solve it. Two test examples: a theoretic mathematical function and a robotic arm control problem were used and for both examples, the EA was able to find a set of suitable solutions that considers multiple trajectories of different adaptation costs.

An different approach [21] presents a multi-swarm Adaptive Solution Chooser (ASC) algorithm for DOPs where switching costs is used. ASC tracks peaks and calculates their fitness variance for individuating, in terms of robustness of the solution, the most reliable ones. The choice between a new solution or keeping the previous solution is decided on multiple factors such as the current solutions fitness values; the fitness value of a prefixed number of higher quality (i.e. fitter) solutions; their switching cost from the current solution.

III. Proposed Modified Moving Peak Benchmark Attractor

In this section we first describe the Modified Moving Peaks Benchmark (MMPB). We then propose our benchmark, Modified Moving Peaks Benchmark Attractor(MMPBA).

A. *Modified Moving Peaks Benchmark*

In order to compare the suitability of MMPBA, the modified version of moving peaks benchmark (MPB) [14, 1, 3] is also used. In MMPB, at each environment change; the height, width and position of each peak randomly changes dependant on a severity factor. This test set is modified to allow each peak to have its own severity factor where it is possible to change some areas of the environment space more severely than others, making this an appropriate benchmark to test ROOT algorithms. The base equation of the Moving Peak Benchmark is described in Eq.(1).

$$F_t(\vec{x}) = \max_{i=1,2,\ldots,m} \left\{ H_t^i - W_t^i * \left\| \vec{X} - \vec{C}_t^i \right\|_2 \right\} \tag{1}$$

where m is the number of peaks, $\vec{x}$ is a solution in the problem space, and H_t^i, W_t^i, C_t^i are the height, width and centre of the ith peak in the tth environment. The MPB is modified by introducing a movement severity for the height, width, centre and shift respectively and is changed as follows:

$$H_{t+1}^i = H_t^i + \text{ height severity }^i * N(0,1) \tag{2}$$

$$W_{t+1}^i = W_t^i + \text{ width severity }^i * N(0,1) \tag{3}$$

$$\vec{C}_{t+1}^i = \vec{C}_t^i + \vec{V}_{t+1}^i \tag{4}$$

where:

$$\vec{V}_{t+1}^i = \text{ shift severity } * \frac{(1-\lambda) * \vec{r} + \lambda * \vec{V}_t^i}{\left\|(1-\lambda) * \vec{r} + \lambda * \vec{V}_t^i\right\|} \tag{5}$$

where $N(0,1)$ is a random number from a gaussian distribution with a zero mean and one variance. The parameter settings of the MMPB are shown in Table II.

B. Modified Moving Peaks Benchmark Attractor

The peaks of the MMPB undergo a random walk due to a random number drawn from a gaussian distribution. The problem may not be predicted accurately and could lead the ROOT framework to fall under prediction-deception [15]. A benchmark problem derived from the MMPB, called MMPB-Attractor (MMPBA) is proposed to allow for the ROOT framework to not fall under prediction-deception.

The benchmark problem uses a random location for each peak in the environment space to act as point at which the peak is attracted to. As the peaks change, the movement is influenced towards the attractor location for which such movement may be able to be predicted by the framework. The benchmark takes the euclidean distance between the optima and its attractor location and and applies this value to the shift factor of the optima. A weighting is used to adjust the severity of the how much the peaks move towards their respective attractor location. A high weighting attracts the peaks more compared to a low weight, allowing a peak to undergo a more severe random walk. In this study, the peaks attractor location does not change for the duration of the experiment. Fig 1 shows the distribution of peak changes in 1000 runs, highlighting the effect of changing the weighting value. For these figures the MMPBA benchmark problem is formed with one peak, the starting location is at [25, 25] and the attractor location is [40, 40]. The peak is allowed to change 100 times, and at each change the location is recorded. Fig 1(a) shows a w_a of 0, which is the original MMPB problem, and highlights that over time the peaks change forms a gaussian distribution around the starting location, therefore for a number of runs in a ROOT experiment the peaks movement cannot be effectively predicted. In Fig 1(c) and Fig 1(b) the weighting value is set to 0.01 and 0.005 respectively. The distribution of the peaks shows the general attraction to the attractor location, in Fig 1(b), the distribution is more sparse with a mean located between the original starting location and attractor location. Such sparsity means that the peak undergoes a random walk while also attracted to the attractor location.

The attractor method can be extended by moving the attractor location itself during every environment change. Such change can be pre-planned in order to produce cyclic, recurrent or periodical changes, or fully stochastic. However, an issue with prediction arises when the training data for the forecasting method within the ROOT framework is dealing with a static attractor problem. If the attractor changes to a new location, the historical data may lead to prediction-deception.

Extending Eq.(5) we can suggest a new peak shift as follows in Eq.(6):

$$\vec{V}_{t+1}^i = \mathrm{S}_s \frac{(1-\lambda)\vec{r} + \lambda\vec{V}_t^i}{\left\|(1-\lambda)\vec{r} + \lambda * \vec{V}_t^i\right\|} + w_a|\vec{A}_t^i - \vec{C}_t^i| \tag{6}$$

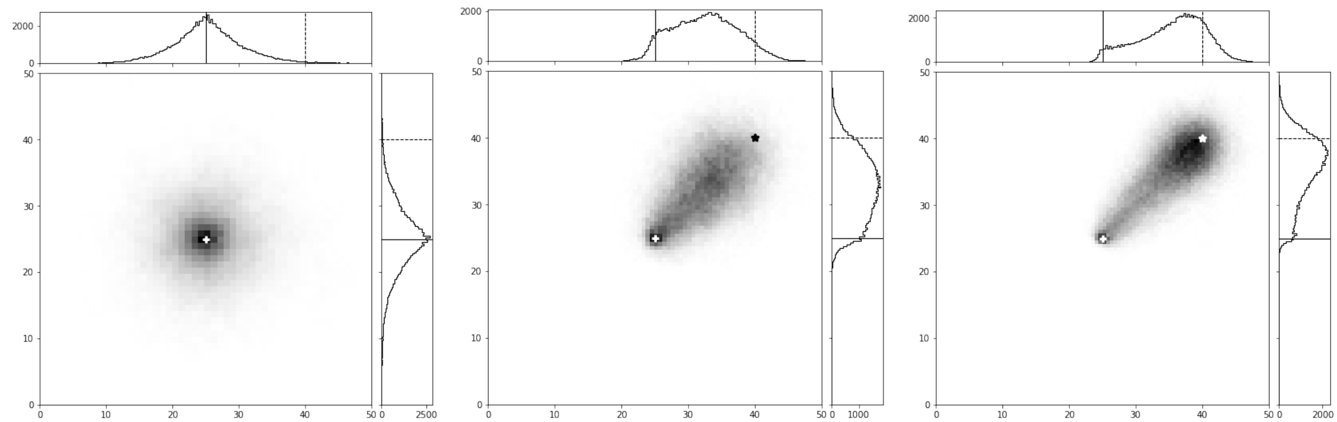
where S_s is the shift severity factor, A_t^i is the attractor location, and w_a the weighting of the attractor. The Euclidean Distance between the attractor and peak is used as the metric to influence the peaks movement.

IV. Experimental Study

A. Methodology

We use the framework proposed by Jin et.al. [9] as our basis for the algorithm, but we significantly modify it to have a system of using a Radial Basis Function, trained on a set of individuals to find an approximate solution representing the current individual, the true fitness values of individuals from historical environments to isolate the effects of the predictor, thus preventing the introduction of approximation errors present in the original framework. A algorithm [22] is used as the optimizer where the parameters can be found in Table I, any population-based optimizer may be used. We also employ a simple restart strategy by keeping the best global particle from the previous environment change and insert it into the new population, therefore the particles are attracted to the last known best position.

Relevant time series forecasting methods are used to test fitness values predictions in the ROOT framework. Four time series regression techniques are considered as the prediction method: Linear and Quadratic Regression [10], a Autoregressive model [9], and Support Vector Regression (SVR) [11]. Comparison with the Linear and Quadratic Regression is to identify the performance of the framework given a simple regression technique. The Autoregressive model is the same prediction method used in the ROOT framework from [9], the parameters for the autoregressive model is unchanged allowing us to compare other prediction methods to the standard method. Finally, a more advanced method, i.e. the SVR approach, is also used to deal with noisy training data.



(a) w_a = 0, creating the original MMPB problem. Normal distribution shows peak is not influenced in a certain direction.

(b) w_a = 0.0005, small weighting in attractor movement creates a weak movement towards the attractor. Distribution is moderately skewed towards the attractor location. Wide distribution in the density chart suggests peaks still undergo a large random walk.

(c) w_a = 0.01, large weighting towards the attractor causes peaks to quickly move to the attractor location. Distribution is strongly skewed towards the attractor location. Higher frequency of peaks movement around the attractor location suggests peak quickly converges to the attractor and then random walks around it for the rest of the environment changes.

Fig. 1. Two-dimensional density distribution and frequency distribution charts of MMPBA peak movement over 100 environment changes for three different values of $w_a \in \{0, 0.005, 0.01\}$. Peak starting location of (25, 25) and attractor location at (40, 40), and represented as solid line and dashed line on frequency distribution chart respectively.

B. Performance Measures

The following performance measure takes into account the chosen solutions fitness performance. Here we take the Average Fitness and Average Survival Time [8] which quantifies the best fitness of the chosen solution in all environments. The performance measures of a ROOT solution can be derived using Eq. (7) and by in Eq. (8):

$$\text{Average Fitness} = \frac{1}{N}\sum_{i=1}^{N} F(i) \tag{7}$$

$$\text{Average Survival Time} = \frac{1}{N}\sum_{i=1}^{N} S(i) \tag{8}$$

where N is the total number of environment changes, $F(i)$ is the fitness of the best individual found for each environment change. $S(i)$ is the survival time of this best individual found at the ith time-varying moment.

TABLE I
PARAMETER SETTINGS FOR PSO AND PREDICTION METHODS

Parameter	Value
r1, r2	2.05
c	0.729
Particle Count	50

V. EXPERIMENTAL RESULTS

Table III shows the average fitness of solutions using the MMPB problem with different dimensions and peaks, comparing different forecasting methods with varying lengths of historical data and the number of future environments to predict. The results show that the all methods achieve the same results given different peaks and dimensions, however linear regression is slightly superior than the other methods, which is also reflected in Table IV giving the results of the average survival time on the MMPB problem with varying peaks, dimensions, and forecasting methods given their parameters.

TABLE II
PARAMETER SETTINGS FOR MMPB AND MMPBA BENCHMARK PROBLEMS

Parameter	Value
Number of Peaks	5, 10, 20
Change Frequency	2500
Shift Severity	Randomized in [0.5, 3]
Height Severity	Randomized in [1, 15]
Width Severity	Randomized in [0.1, 1.5]
Peaks Shape	Cone
Number of Dimensions	2, 5, 10
Correlations Coefficient	0
Peaks Location Range	[0, 50]
Peak Height	[30, 70]
Peak Width	[1, 12]
Initial Height Value	50
Initial Width Value	6
Number of Environments	100
Historical Length, p	$[1, 2, 3]$
Prediction Length, q	$[4, 6, 8]$
Survival Threshold, V	$[40, 45, 50]$
Maximum Survival Length	10
Other Parameters	MMPBA Attractor Weighting, w_a = $[0, 0.005, 0.01]$

The length of the historical data and prediction shows that it is more appropriate to use a shorter length for both, with a increase of average fitness for most forecasting methods

TABLE III

AVERAGE FITNESS RESULTS OF DIFFERENT PREDICTION METHODS ON MMPB, WITH A VARIED NUMBER OF PEAKS AND DIMENSIONS, AND LENGTH HISTORICAL AND PREDICTED FITNESS SOLUTIONS.

		Dimensions = 2			Dimensions = 5			Dimensions = 10		
(p, q)	Algorithm	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20
(1, 4)	Linear Regression	23.50	27.06	32.43	17.75	23.65	27.77	15.61	18.27	23.32
	Quadratic	23.11	26.72	32.24	17.74	23.63	27.68	15.48	17.61	23.11
	AR	22.27	26.31	32.27	16.85	23.23	27.34	15.23	17.64	22.47
	SVR	22.53	26.16	32.00	16.85	23.25	27.94	15.64	17.56	22.22
(2, 6)	Linear Regression	22.36	32.59	39.61	16.97	29.02	34.17	12.56	21.47	34.63
	Quadratic	22.69	31.51	37.93	16.54	29.18	34.05	11.92	21.88	32.54
	AR	22.22	31.93	37.22	16.12	29.03	33.43	11.86	22.25	32.53
	SVR	22.62	31.15	37.12	16.69	29.50	33.97	11.95	21.70	33.22
(3, 8)	Linear Regression	23.62	37.09	39.73	20.99	34.60	35.19	16.59	33.16	28.63
	Quadratic	22.12	36.98	37.89	19.42	33.90	34.15	15.51	32.67	27.43
	AR	22.31	36.91	37.43	19.59	34.53	33.78	15.63	32.83	27.33
	SVR	22.26	37.02	38.19	19.57	34.87	32.79	15.48	32.67	27.18

TABLE IV

AVERAGE SURVIVAL TIME RESULTS OF DIFFERENT PREDICTION METHODS ON MMPB, WITH A VARIED NUMBER OF PEAKS AND DIMENSIONS, AND LENGTH HISTORICAL AND PREDICTED FITNESS SOLUTIONS.

		Dimensions = 2			Dimensions = 5			Dimensions = 10		
V	Algorithm	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20
40	Linear Regression	4.37	5.63	6.30	1.07	2.28	3.10	0.79	1.42	1.66
	Quadratic	4.44	5.55	6.26	1.07	2.29	3.08	0.73	1.41	1.64
	AR	4.36	5.60	6.29	1.06	2.29	3.05	0.81	1.37	1.64
	SVR	4.40	5.56	6.23	1.02	2.37	3.03	0.77	1.38	1.69
45	Linear Regression	3.65	4.67	5.56	0.29	0.73	0.87	0.20	0.33	0.80
	Quadratic	3.58	4.67	5.63	0.35	0.65	0.83	0.19	0.27	0.77
	AR	3.64	4.67	5.55	0.29	0.64	0.87	0.18	0.24	0.76
	SVR	3.56	4.87	5.49	0.20	0.69	0.80	0.19	0.24	0.75
50	Linear Regression	2.27	3.27	4.17	0.08	0.42	0.77	0.09	0.19	0.30
	Quadratic	2.23	3.21	4.15	0.12	0.38	0.75	0.07	0.07	0.24
	AR	2.26	3.20	4.17	0.10	0.39	0.75	0.09	0.15	0.27
	SVR	2.18	3.14	4.18	0.09	0.43	0.71	0.08	0.15	0.31

TABLE V

AVERAGE FITNESS RESULTS OF DIFFERENT PREDICTION METHODS ON MMPBA, WITH A VARIED NUMBER OF PEAKS AND DIMENSIONS, AND LENGTH HISTORICAL AND PREDICTED FITNESS SOLUTIONS.

		Dimensions = 2			Dimensions = 5			Dimensions = 10		
(p, q)	Algorithm	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20
(1, 4)	Linear Regression	27.40	31.05	32.81	22.12	26.58	31.79	19.21	20.55	25.28
	Quadratic	27.33	28.94	32.54	21.80	28.41	30.64	18.77	19.31	24.45
	AR	26.58	29.19	31.66	23.01	27.11	30.45	18.74	20.21	24.94
	SVR	26.30	29.16	32.00	21.84	27.83	30.90	19.11	19.20	23.01
(2, 6)	Linear Regression	25.83	34.33	40.54	20.27	30.67	35.60	14.33	26.30	36.61
	Quadratic	23.80	34.09	39.84	20.09	30.34	35.83	13.99	24.90	36.45
	AR	24.34	34.13	39.84	19.66	31.24	36.31	14.38	25.38	36.63
	SVR	24.32	34.07	39.92	19.92	31.11	35.71	14.41	25.63	36.52
(3, 8)	Linear Regression	25.56	39.01	42.02	24.81	37.89	36.06	17.02	27.17	28.37
	Quadratic	25.04	39.16	41.93	22.05	37.20	36.29	18.30	27.05	27.58
	AR	25.07	39.28	41.70	22.90	37.30	36.53	18.82	26.77	28.04
	SVR	25.81	39.16	41.18	22.60	36.33	35.50	18.43	26.63	27.90

compared to longer lengths of historical data and predictions.

The average survival time results in Table IV using the MMPB problem with varying peaks and dimensions, and different levels of thresholds also exhibit the same results of Table VI. There is no clear difference between the forecasting methods, and the higher the survival threshold the lower the average survival time.

For both the average fitness and average survival time results the conclusion is the same with comparing different forecasting methods and it suggests that the metrics are unsuitable in finding ROOT solutions if the accuracy of the prediction is not known or the performance measures used are not suitable if this accuracy is considered between forecasting methods as proved in [8]. In this study, the linearity of the forecasting methods has a direct impact in both metrics. Considering the survival time metric with a linear regression forecasting method, if the historical data is continuously above the survival threshold with a small variance, then the subsequent predicted

TABLE VI
AVERAGE SURVIVAL TIME RESULTS OF DIFFERENT PREDICTION METHODS ON MMPBA, WITH A VARIED NUMBER OF PEAKS AND DIMENSIONS, AND LENGTH HISTORICAL AND PREDICTED FITNESS SOLUTIONS.

		Dimensions = 2			Dimensions = 5			Dimensions = 10		
V	Algorithm	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20
40	Linear Regression	5.52	6.41	6.40	1.52	2.81	4.08	1.73	2.22	2.54
	Quadratic	4.71	6.42	6.37	1.84	2.30	3.92	0.40	1.62	1.62
	AR	4.46	6.03	6.71	1.74	2.35	3.20	1.59	1.58	2.31
	SVR	5.01	6.08	6.86	1.61	2.60	3.42	1.34	2.49	2.18
45	Linear Regression	4.07	5.16	5.94	1.13	0.95	1.40	1.05	0.99	1.00
	Quadratic	3.81	5.00	5.70	0.75	1.03	1.08	0.48	0.50	0.84
	AR	4.26	5.15	5.85	1.07	0.47	1.24	1.38	0.76	1.55
	SVR	4.16	4.91	6.00	0.62	1.07	1.31	1.12	0.57	1.37
50	Linear Regression	3.32	3.44	4.85	0.81	0.54	1.99	1.03	0.79	1.23
	Quadratic	2.98	3.32	4.31	0.68	1.03	1.00	0.23	0.42	0.71
	AR	2.92	3.99	4.30	1.43	1.07	1.73	0.54	0.46	0.69
	SVR	2.98	3.88	4.59	0.33	1.36	1.72	0.38	0.42	0.41

TABLE VII
AVERAGE FITNESS RESULTS OF DIFFERENT ATTRACTOR WEIGHTS FOR MMPBA, WITH A VARIED NUMBER OF PEAKS AND DIMENSIONS, USING A AR PREDICTION METHOD WHERE P=1, Q=4.

	Dimensions = 2			Dimensions = 5			Dimensions = 10		
W	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20	P = 5	P = 10	P = 20
0	22.27	26.31	32.27	16.85	23.23	27.34	15.23	17.64	22.47
0.005	24.89	28.46	33.48	19.86	25.04	28.53	17.40	20.25	23.80
0.1	0.37	4.37	19.22	-0.02	8.54	12.53	-3.62	1.38	6.65

results are above this threshold until the maximum length is reached. If the historical data has a larger variance, and as the forecasting method is linear, the predicted values may fall below the survival threshold within the first few survival lengths or continue until the maximum length, influencing the results to show a higher average survival length. With non-linear functions the accuracy may be reduced leading to more appropriate results with less extreme survival lengths. In this case, averaging extreme results and good results achieves the same result.

Comparisons with Table III and Table V show the use of the standard MMPB problem and the problem proposed in this work, MMPBA. Results show that MMPBA has a slight higher average fitness in most cases reflecting the suitability of a predictable problem in ROOT studies.

Table VII shows the effect of different weightings of the attractor function in MMPBA using AR as the predictor and p=1, q=4. A smaller weighting suggests that the peak is attracted less to the attractor location and undergoes more random walk therefore it cannot be predicted effectively, whereas a higher weighting attracts the peak more severely leading to a heuristic that can be predicted. With a weighting of 0 the benchmark is the same as the standard MMPB problem. The results show that a small attraction weighting leads to slightly higher average fitness than that of a higher attraction weight and a weighting of 0.

One reason for this is that a heuristic that creates a predictable environment space is ideal for the ROOT framework - the ROOT framework does not predict the movement of the peaks rather the estimation of the individuals fitness. A prediction method's accuracy may be higher if the uncertainty of the individuals fitness in future environments is reduced by reducing the random walk of the peak. This forecasting error, and that using the standard MMPB problem, is further reduced if the linearity of the prediction method matches that of the peak shape, i.e. a simple linear regression method is appropriate for a conical peak shape.

Worse results in higher attractiveness weightings suggests that the movement of the peak between changes is too great for the ROOT metric, in this case the average fitness, and dependant on the peak shape. With a small peak width and large movement the fitness drops significantly in future environment changes. Methods to find a robust solution considering this characteristic may need more research.

VI. CONCLUSION AND FUTURE DIRECTION

In this paper we investigate the characteristics of different time series forecasting methods for the predictor component in the ROOT framework proposed in [9]. The framework were tested on the MMPB function, and a new benchmark problem extending MMPB called MMPB-Attractor was proposed. This benchmark problem guides the peaks movement towards a specific position in the environment increasing the predictability of the problem.

The findings of the paper were that given the performance measures and metrics used there is not much difference in choosing a simple Linear Regression to more advanced prediction methods. These methods may achieve a more accurate result, however this may be masked by a simple method were the predicted future fitness values are a significantly higher but not accurate - unless using the updated ROOT metrics that incorporate prediction error.

Benchmark problems that cannot be predicted will deceive the optimizer and framework as the peaks may move using a random walk whereas, a benchmark problem that includes a predictable heuristic leads to slightly better results as shown using MMPBA. A characteristic that considers the distance a peak moves between changes was highlighted showing that the ROOT framework does not work well.

Future work will explore this issue and include uncertainty in the approximation or prediction of the future possible solution. The relationship of quick recovery and requirement satisfaction in ROOT problems will be explored. Finally, MMPBA will be extended to include cyclical, recurrent, and periodic predictive factors in order to identify ROOT algorithms that can adapt to predictable changes.

REFERENCES

[1] V. S. Aragón and S. C. Esquivel, "An evolutionary algorithm to track changes of optimum value locations in dynamic environments," *Journal of Computer Science and Technology*, vol. 4, no. 03, pp. p. 127–133, Oct. 2004.

[2] Y. Jin, J. Branke *et al.*, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on evolutionary computation*, vol. 9, no. 3, pp. 303–317, 2005.

[3] C. Cruz, J. R. González, and D. A. Pelta, "Optimization in dynamic environments: a survey on problems, methods and measures," *Soft Computing*, vol. 15, no. 7, pp. 1427–1448, 2011.

[4] S. Yang, "Evolutionary computation for dynamic optimization problems," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO Companion '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 629–649.

[5] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 832–846.

[6] J. J. Grefenstette *et al.*, "Genetic algorithms for changing environments," in *PPSN*, vol. 2, 1992, pp. 137–144.

[7] X. Yu, Y. Jin, K. Tang, and X. Yao, "Robust optimization over time — a new perspective on dynamic optimization problems," in *IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–6.

[8] H. Fu, B. Sendhoff, K. Tang, and X. Yao, "Finding robust solutions to dynamic optimization problems," in *Proceedings of the 16th European Conference on Applications of Evolutionary Computation*, ser. EvoApplications'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 616–625.

[9] Y. Jin, K. Tang, X. Yu, B. Sendhoff, and X. Yao, "A framework for finding robust optimal solutions over time," *Memetic Computing*, vol. 5, no. 1, pp. 3–18, Mar 2013.

[10] T. K. Keyes, "Applied regression analysis and multivariable methods," *Technometrics*, vol. 43, no. 1, pp. 101–101, 2001.

[11] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[12] H. Fu, B. Sendhoff, K. Tang, and X. Yao, "Robust optimization over time: Problem difficulties and benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 731–745, Oct 2015.

[13] Haobo Fu, B. Sendhoff, K. Tang, and X. Yao, "Characterizing environmental changes in robust optimization over time," in *2012 IEEE Congress on Evolutionary Computation*, June 2012, pp. 1–8.

[14] P. Novoa-Hernández, D. A. Pelta, and C. C. Corona, "Approximation models in robust optimization over time - an experimental study," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–6.

[15] D. Yazdani, T. T. Nguyen, J. Branke, and J. Wang, "A new multi-swarm particle swarm optimization for robust optimization over time," in *Applications of Evolutionary Computation*, G. Squillero and K. Sim, Eds. Cham: Springer International Publishing, 2017, pp. 99–109.

[16] D. Yazdani, T. T. Nguyen, and J. Branke, "Robust optimization over time by learning problem space characteristics," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 143–155, Feb 2019.

[17] Y. Guo, M. Chen, H. Fu, and Y. Liu, "Find robust solutions over time by two-layer multi-objective optimization method," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, July 2014, pp. 1528–1535.

[18] Y. Huang, Y. Ding, K. Hao, and Y. Jin, "A multi-objective approach to robust optimization over time considering switching cost," *Inf. Sci.*, vol. 394, no. C, pp. 183–197, Jul. 2017.

[19] D. Yazdani, T. T. Nguyen, J. Branke, and J. Wang, "A multi-objective time-linkage approach for dynamic optimization problems with previous-solution displacement restriction," in *Applications of Evolutionary Computation*, K. Sim and P. Kaufmann, Eds. Cham: Springer International Publishing, 2018, pp. 864–878.

[20] S. Salomon, G. Avigad, P. J. Fleming, and R. C. Purshouse, "Optimization of adaptation - a multi-objective approach for optimizing changes to design parameters," in *Evolutionary Multi-Criterion Optimization*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 21–35.

[21] D. Yazdani, J. Branke, M. N. Omidvar, T. T. Nguyen, and X. Yao, "Changing or keeping solutions in dynamic optimization problems with switching costs," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '18. New York, NY, USA: ACM, 2018, pp. 1095–1102.

[22] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, vol. 4. Citeseer, 1995, pp. 1942–1948.